

# Optimally fuzzy scale-free memory

Karthik H. Shankar\* and Marc W. Howard

*Center for Memory and Brain, Boston University*

Any system with the ability to learn from a time series and predict the future must have a memory representing the information from the recent past. In cases where the external environment generating the time series has a fixed scale, the memory can be a simple shift register—a moving window of finite width extending into the past. The width of the window should be large enough to describe the largest scale relevant for predicting the signal. However, such a traditional buffer is inappropriate if the longest relevant scale is not known *a priori*, or if the signal has structure at many different time scales. It is well known that signals with scale-free long range correlations are found in many physical environments. Hence we argue in favor of a memory that is a scale-free fuzzy buffer which implicitly accounts for scale-free fluctuations in naturally generated signals. Based on a neuro-cognitive model of internal time, we construct a fuzzy buffer that optimally sacrifices the accuracy of information representation in order to represent exponentially long time scales without an explosion in capacity demands. Using several illustrative time series we demonstrate the advantage of the fuzzy buffer over the shift register in time series forecasting. We suggest that this method for representing time-varying signals may be of broad utility in a variety of applications.

## I. INTRODUCTION

Time series forecasting is a generic problem that arises in many contexts ranging from understanding the occurrence of solar flares to understanding stock market fluctuations. A basic question that arises with respect to forecasting is, how much of the recent past of the time series is required to generate accurate predictions for the future? If there are no significant correlations beyond a particular scale, then a *shift register* [1] of appropriate size should suffice to accurately hold the information from the recent past leading up to any moment. However, when the relevant statistics are unknown, it would be disadvantageous to subscribe to a fixed size shift register. There are many instances where we might be concerned that the relevant information needed to forecast the time series could be spread over very long timescales. For example, complex interconnected dynamical systems often exhibit scale-free long range correlations in their spatial and temporal fluctuations, commonly known as  $1/f$  fluctuations. Such long range correlations can be found in statistics of natural images [2–4], speech and music [5], brain activity [6], economics [7] and even in human cognition [8–10]. Though the physical generating mechanism underlying such long range correlations is an active subject of debate among engineers and physicists [11], our interest here is to simply point out its ubiquitous existence with a focus on the following question: If an intelligent learner is to learn using the time series generated by real world complex systems which may have correlation over very long delays, is there an optimal way to represent the time series in the memory buffer of the learner?

Here we argue that it is advantageous for a buffer with finite resources to reflect the natural scale-free temporal structure associated with the uncertainties of the world. If one were to *a priori* assume that the time series is generated by a system with long range correlations then an event that happened 100 seconds ago does not have to be represented as accurately in time as an event that happened 10 seconds ago. By sacrificing the accuracy in a scale-free fashion, the learner can

---

\* shankark@bu.edu

optimally gather the relevant statistics from the time series with a built-in assumption that the series exhibits long range correlations. In this paper, we describe such a scale-free fuzzy buffer and discuss its advantages over a shift register in extracting statistics and forecasting time series from generic external environment.

Of course, representing the recent history in an optimal fashion is not sufficient to successfully predict the future time series. It is crucial to learn the relevant statistics with an efficient learning algorithm. When the processes generating the time series is unknown or highly complex, even simple statistical learning methods such as correlation and spectral analysis can extract the significant statistics underlying the time series. Though a variety of sophisticated machine learning algorithms exist (see e.g., [12, 13]), there is none that is constructed to act on a fuzzy memory buffer to extract the relevant statistics. The choice of the learning algorithm is modular to the choice of the memory buffer. The focus of this paper is the memory buffer and not on the learning algorithm *per se*; in section 4 we use a simple linear regression algorithm to demonstrate the utility of the fuzzy buffer in time series forecasting.

The layout of the paper is as follows. In section 2 we start with a mathematical motivation for capacity-accuracy tradeoff in the memory buffer based on some general properties of long-range correlated time series. We explain the criteria for optimally sacrificing accuracy of information representation in the memory buffer for the sake of capacity to represent longer time scales. In section 3 we describe a specific method for representing temporal history of the time series in a scale-free way based on a neuro-cognitive model of internal time, TILT [14]. Mathematically, this method is equivalent to encoding the Laplace transform of the time series and approximating its inverse to reconstruct a fuzzy representation of the time series. We then construct the fuzzy buffer by imposing the criteria of optimally sacrificing the accuracy of information representation, and show that with limited resources the fuzzy buffer has the capacity to represent exponentially larger timescales in comparison to a shift register. In section 4, we compare the performance of the fuzzy buffer and the shift register in time series forecasting. Using an artificially generated long-range correlated time series and empirically observed time series for sunspots and Earth's temperature, we show that the fuzzy buffer consistently outperforms the shift register in time series forecasting. Finally, we conclude by pointing out that adopting such a fuzzy buffer as a baseline memory representation in statistical learning models could be very useful.

## II. MOTIVATION FOR CAPACITY-ACCURACY TRADEOFF

Let us assume that the learner must learn and forecast a real valued time series that has a two point correlation function that falls off like a power law. Naturally occurring time series will most certainly contain more subtle features like higher order correlations, but they are currently irrelevant for motivating the need for a fuzzy buffer. Hence for simplicity, let  $v_\tau$  represent a stationary time series indexed by time stamp  $\tau$ , which has a zero mean, finite variance, and power-law two point correlations, namely  $\langle v_\tau v_{\tau'} \rangle \simeq 1/|\tau - \tau'|^\alpha$ , for large temporal differences  $|\tau - \tau'|$ . When  $\alpha \leq 1$ , the time series is said to possess long range correlations [15]. Our aim here is to simply represent this time series in a memory buffer so as to optimally extract its statistical properties and forecast the future values of the time series. For this purpose, it is useful to view the time series from the perspective of it being generated by a generic statistical algorithm, the ARFIMA model [10, 16, 17]. The basic idea behind this algorithm is that white noise at each time step can be fractionally integrated to generate a time series with long range correlations. It turns out that the time series can be viewed as generated by an infinite auto-regressive generating function integrating white noise. Without loss of generality, consider the current time step in the time series to be  $\tau = 0$ , and let the time steps be uniformly spaced so that they can be simply

labeled by integers. The value  $v_o$  at the current time step is a linear combination of white noise  $\eta_o$  and the values  $v_n$ s from past times  $\tau = n$ .

$$v_o = \eta_o + \sum_{n=1}^{\infty} a(n)v_n. \quad (1)$$

The ARFIMA model uniquely specifies the regression coefficients  $a(n)$  in terms of the exponent  $\alpha$ .

$$a(n) = \frac{(-1)^{n+1}\Gamma(d+1)}{\Gamma(n+1)\Gamma(d-n+1)}, \quad (2)$$

where  $d$  is the fractional integration power given by  $d = (1 - \alpha)/2$ . It is known that an ARFIMA time series is stationary and long range correlated with finite variance only when  $d \in (0, 1/2)$  or  $\alpha \in (0, 1)$  [16, 17]. The asymptotic behavior of  $a(n)$  for large  $n$  can be obtained by applying Euler's reflection formula and Stirling's formula to approximate the Gamma functions in eq. 2.

$$\lim_{n \gg 1} a(n) = \left[ \frac{\Gamma(d+1) \sin(\pi d)}{\pi} \right] n^{-(1+d)}. \quad (3)$$

The purpose of writing out eq. 1 is to simply note that the coefficient  $a(n)$  can be interpreted as a measure of the relevance of  $v_n$  in predicting  $v_o$ . We shall use this interpretation to motivate an optimal way to represent the  $v_n$ s in a memory buffer. If the buffer has unlimited storage resources, then all  $v_n$ s can be represented with perfect accuracy, with a unique buffer node for each  $n$ . At each time step, the value in the  $n$ -th node of the buffer can be shifted to the  $n+1$ -th node and the value  $v_o$  can be filled into the first node. This memory buffer is a shift register of infinite size, and it ensures a perfect representation of the past time series at each moment. Now the question we address here is, if the buffer has only finite storage resources, is there a way to optimally sacrifice the accuracy in order to represent long time scales? Since the relative importance of  $v_n$  reduces with increasing  $n$  (as seen from eq. 2), we propose that the ideal buffer should store a weighted average of  $v_n$ s over monotonically increasing bin-sizes such that the information overlap between successive bins is a constant. Figure 1 represents this idea schematically. We motivate the construction of the ideal buffer based on the principle that both (i) the error induced by averaging and (ii) the information redundancy induced by averaging should be equally distributed over all scales that are represented.

From fig. 1, note that in a shift register (SR) the value  $v_n$  will be stored in, and only in, the  $n$ -th node of the buffer. Let us now consider a smeared shift register (SSR) where  $v_n$  is not only stored in the  $n$ -th node of the buffer, but is smeared across the nodes around the  $n$ -th node. The purpose of smearing is to acknowledge the existence of fluctuations inherent to the generation of the time series. The utility of smearing is best explained in the context of a binary valued time series corresponding to the occurrence of a stimulus ( $v_n=1$ ) or not ( $v_n=0$ ) at each time step  $n$ . The stimulus represented by the series could be anything, like the occurrence of solar flares, or occurrence of lightning in a stormy night, or even occurrence of an economic depression. Let us say that the learner somehow makes an association that the occurrence of the stimulus at the  $m$ -th time step in the past it is a strong predictor of the current re-occurrence of the stimulus. If the learner acquired such a statistic based on only a few learning instantiations, then it would be advantageous for the learner to expect some fluctuations in the number  $m$  to account for natural fluctuations that exist in the generation of the time series. If the learner generalizes the learned statistic to values around the  $m$ -th time step, then while forecasting the future the learner will not just expect the re-occurrence of the stimulus exactly  $m$  time steps in the future of a prior occurrence, but will expect the re-occurrence in a spread-out fashion around  $m$ -th time step in the

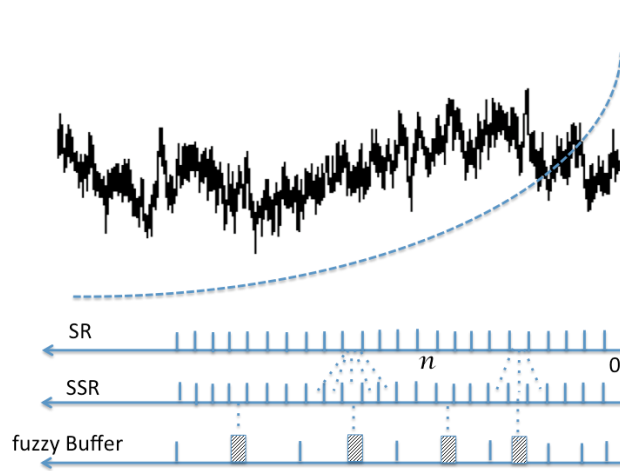


FIG. 1. A sample time series  $\{v_n\}$  with power-law two-point correlation is plotted w.r.t time( $n$ ). The current time step is taken to be  $n = 0$ , and each  $v_n$  represents the value at the  $n$ -th time step in the past. The dotted curve shows  $a(n)$  (see eq. 1), the relative importance of  $v_n$  in predicting  $v_o$ . The upper  $x$ -axis represents the shift register (SR) which stores each  $v_n$  in a unique node. After each time step, the value stored in each node is shifted to the next node in the left, and the value in the last node is ejected. The middle  $x$ -axis represents a smeared shift register (SSR), where each node stores the weighted sum over a bin of SR nodes. The lower  $x$ -axis represents the optimally fuzzy buffer which is essentially a collection of a subset of SSR nodes. The bins surrounding each fuzzy buffer node indicates the range of SR nodes involved in its construction. The size of the bins and overlap between bins can be optimally chosen to reflect the behavior of  $a(n)$ .

future. This can happen if the relevance of  $v_m$  in predicting  $v_o$  is shared by nodes surrounding the  $m$ -th node, which in turn can be achieved by smearing or re-distributing the value  $v_m$  into nodes around the  $m$ -th node. Since the learner is unaware of the statistics underlying the time series prior to learning it, neither the value of  $m$  nor the spread in the fluctuations can be *a priori* guessed by the learner. But given the ubiquity of scale free fluctuations in naturally occurring time series, we argue that the best strategy for the learner is to represent the information from every time step in the past in a smeared fashion and require the smearing to be evenly spread across all timescales. Such a representation of the time series in the memory buffer is schematically shown as the smeared shift register (SSR) in the middle  $x$  axis of fig. 1.

As shown in fig. 1, the value in each SR node is distributed onto a set of neighboring SSR nodes. In effect, each SSR node essentially encodes a weighted sum over a set of SR nodes, which we shall refer to as a bin. To consider the effect of such smearing on the prediction of  $v_o$ , note from eq. 1 that taking a weighted sum of  $v_n$ s over all the nodes in a bin is equivalent to treating  $a(n)$  to be a constant over the bin. If  $\Delta_n$  is the size of the bin, then  $\Delta_n da(n)/dn$  is a measure of the error induced in the prediction due to smearing. To estimate the optimal size of the bins, we use a simple guiding principle that the contribution of smearing to the error in prediction should be proportional to its contribution to the prediction itself. That is,

$$\Delta_n \frac{da(n)}{dn} \propto a(n) \quad (4)$$

This principle ensures, at least heuristically, that the smear-induced error is equally distributed over all scales relevant to the prediction. Since  $a(n)$  shows a power-law behavior for large  $n$ , this principle yields  $\Delta_n \propto n$ .

If we had unlimited storage resources, a unique SSR node could be assigned for each  $n$ , representing the weighted sum over the bin of SR nodes centered around that  $n$ . But note that for large  $n$ , the bins corresponding to successive SSR nodes will be highly overlapping and most of the information represented by those nodes will be redundant. In a realistic situation with limited resources allocated for the buffer, we could make a smarter choice by representing only a subset of the SSR nodes in the buffer, thereby minimizing the information overlap between neighboring nodes while representing longer time scales in the buffer. To optimally pick out the SSR nodes that should be included in the buffer, we require the information overlap between neighboring nodes at all scales to be a constant. This principle ensures that the information redundancy is equally distributed over all scales relevant to the prediction. For motivational purpose, consider the simplest case where the nodes are chosen such that the bins are non-overlapping and precisely tile the entire timeline. In this case, there is zero information overlap between neighboring nodes. Since we have argued in the previous paragraph that the size of the bin around the  $n$ -th SSR node should be proportional to  $n$ , the maximum time scale that can be represented by the buffer will be related to the exponential of the total number of nodes in the buffer. In comparison to a shift register where the number of nodes in the buffer is directly proportional to the longest time scale represented, this fuzzy buffer can represent information from much longer timescales.

Although the fuzzy buffer represents the time series with fewer resources than the shift register, it is non-trivial to construct the fuzzy buffer we have just described without explicitly having access to a shift register storing the entire time series. A crucial feature of a memory buffer should be self-sufficiency. That is, the information represented in the buffer should evolve at each time step only from the incoming input and the already stored information in the buffer. However, in the scheme described in figure 1, the SR is needed to construct the SSR which is needed to construct the fuzzy buffer. The information lost in weighted-averaging and discarding the intermediate SSR nodes, are essential in determining the information to be represented by the fuzzy buffer at the subsequent time step. Hence such a construction of the fuzzy buffer is not self sufficient to evolve in time. It is hard to argue that a fuzzy buffer saves resources if one needs to have much more extensive resources to construct it! In the next section, we describe a mathematically elegant construction of a representation of temporal history based on encoding and inverting the Laplace transformation of the time series. This method leads to a fuzzy buffer that is self-sufficient to evolve in time without requiring a shift register for its construction. As such, this method provides an efficient method for storing a scale-invariant representation of history.

### III. CONSTRUCTING A SCALE-FREE FUZZY BUFFER FROM A REPRESENTATION OF STIMULUS HISTORY

In this section we describe a method for constructing the scale-free fuzzy buffer. We begin by describing a mathematical model of psychological time, called TILT [14], developed to account for findings from animal and human behavior. This model gives the mathematical basis for representing the stimulus history in a scale-free fashion with properties of the smeared shift register that was described in the previous section. We then describe several critical considerations necessary to implement the mathematical model into a set of buffer nodes that leads to the optimally fuzzy buffer.

Consider a real valued function  $\mathbf{f}(\tau)$  to generate the time series in real time  $\tau$ . Our aim now is to construct a memory that represents  $\mathbf{f}(\tau)$  leading up to the present moment as activity distributed over a set of nodes. The shift register is a simple solution to this problem. One could construct a shift register from a set of nodes chained back to back such that at each time step the functional value of  $\mathbf{f}$  is transmitted to the first node in the chain and the information from each node is

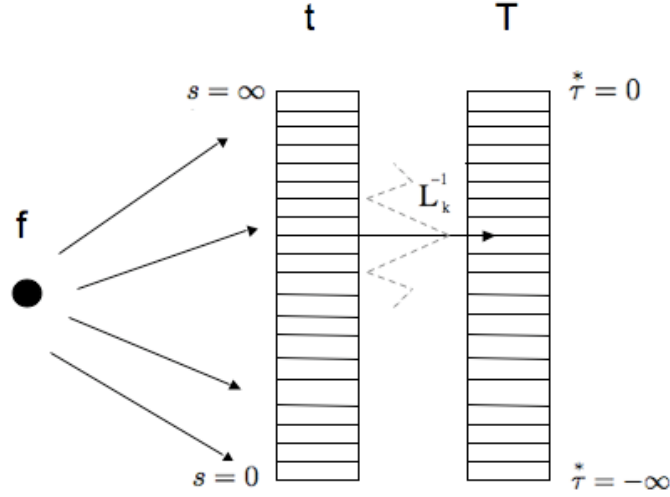


FIG. 2. The scale-free fuzzy representation - Each node in the  $\mathbf{t}$  column is a leaky integrator with a specific decay constant  $s$  that is driven by the functional value  $\mathbf{f}$  at each moment. The activity of the  $\mathbf{t}$  column is transcribed at each moment by the operator  $\mathbf{L}_k^{-1}$  to represent the past functional values in a scale-free fuzzy fashion in the  $\mathbf{T}$  column.

transmitted to the next node downstream. Under these circumstances, each node of the shift register will store the value of  $\mathbf{f}$  from a specific moment in the past. Assuming that there is no error in transmitting from one node to the next and there are an infinite number of nodes, the shift register will accurately hold the entire history up to the present moment.

We will now describe a more sophisticated method to represent history [14]. This method results in a fuzzy estimate of  $\mathbf{f}(\tau)$  using two columns of nodes  $\mathbf{t}$  and  $\mathbf{T}$  as shown in fig. 2. The  $\mathbf{T}$  column estimates  $\mathbf{f}(\tau)$  up to the present moment, while the  $\mathbf{t}$  column is an intermediate step used to construct  $\mathbf{T}$ . The nodes in the  $\mathbf{t}$  column are leaky integrators with decay constants denoted by  $s$ . Each leaky integrator independently gets activated by the value of  $\mathbf{f}$  at any instant and gradually decays according to

$$\frac{d\mathbf{t}(\tau, s)}{d\tau} = -s\mathbf{t}(\tau, s) + \mathbf{f}(\tau). \quad (5)$$

At every instant, the information in the  $\mathbf{t}$  column is transcribed into the  $\mathbf{T}$  column through a linear operator  $\mathbf{L}_k^{-1}$ .

$$\begin{aligned} \mathbf{T}(\tau, \tau^*) &= \frac{(-1)^k}{k!} s^{k+1} \mathbf{t}^{(k)}(\tau, s) : \text{ where } s = -k/\tau^* \\ \mathbf{T} &\equiv \mathbf{L}_k^{-1}[\mathbf{t}]. \end{aligned} \quad (6)$$

Here  $k$  is any positive integer and  $\mathbf{t}^{(k)}(\tau, s)$  is the  $k$ -th derivative of  $\mathbf{t}(\tau, s)$  with respect to  $s$ . The nodes of the  $\mathbf{T}$  column are labeled by the parameter  $\tau^*$  and are in one to one correspondence with the nodes of the  $\mathbf{t}$  column which are labeled by  $s$ . The correspondence between  $s$  and  $\tau^*$  is given by  $s = -k/\tau^*$ . We refer to  $\tau^*$  as the *internal time* because it turns out that at any moment  $\tau$ , a  $\tau^*$  node approximately represents the value of  $\mathbf{f}$  at a time  $\tau + \tau^*$  in the past. The maximum value of  $\tau^*$

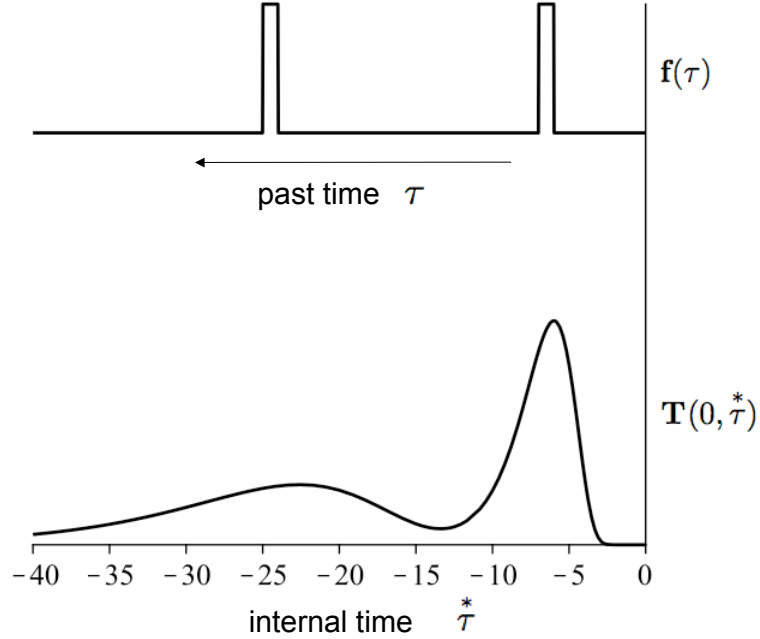


FIG. 3. The function  $\mathbf{f}(\tau)$  is generated by a stimulus presented twice in the recent past. Taking the present moment to be  $\tau = 0$ , the momentary activity distributed across the  $\mathbf{T}$  column nodes is plotted.

can be made as large as needed at the cost of resources, but for mathematical idealization we can take  $\tau^*$  to  $-\infty$ .

The crucial mathematical inspiration of this approach is that  $\mathbf{t}(\tau, s)$  encodes the Laplace transform of the entire history of the function  $\mathbf{f}$  at any moment  $\tau$ , and the operator  $\mathbf{L}_k^{-1}$  approximately inverts the Laplace transform [18], with the approximation being almost perfect for large values of  $k$ . When  $k \rightarrow \infty$ ,  $\mathbf{T}(\tau, \tau^*)$  is a faithful representation of the history of  $\mathbf{f}$  from  $\tau$  to  $-\infty$ , that is  $\mathbf{T}(\tau, \tau^*) \simeq \mathbf{f}(\tau + \tau^*)$  for all values of  $\tau^*$  from 0 to  $-\infty$ . Hence when  $k \rightarrow \infty$ ,  $\mathbf{T}$  behaves exactly like a shift register. Although the result of this computation is identical to a shift register, note that the mechanism is completely different. Information is not transmitted directly from one node in  $\mathbf{T}$  column to the next, as in a shift register. Rather, at each moment, the column of  $\mathbf{t}$  nodes holds information about the entire history of  $\mathbf{f}$  and the  $\mathbf{T}$  column extracts this information to reconstruct the history. This reconstruction is perfect when  $k \rightarrow \infty$ , but is only approximate when  $k$  is finite. It turns out that the error in reconstruction behaves exactly as what we would expect in a smeared shift register described in the previous section. For example, if the current moment is  $\tau = 0$ , then the value of  $\mathbf{f}$  at a particular past moment  $\tau_o$  is accurately represented by the node  $\tau^* = \tau_o$  when  $k \rightarrow \infty$ . But when  $k$  is finite, the value of  $\mathbf{f}$  at the past moment  $\tau_o$  is smeared over a range of  $\tau^*$  nodes. It turns out that this smear is scale invariant and grows linearly with  $\tau_o$ .

To illustrate the behavior of  $\mathbf{T}$  as a smeared shift register, consider the function  $\mathbf{f}(\tau)$  generated by a stimulus that occurred twice in the recent past. With the present moment taken as  $\tau = 0$ , figure 3 shows a function  $\mathbf{f}(\tau)$  that briefly takes non-zero values twice in the past (top), and the estimate of  $\mathbf{f}(\tau)$  present in the  $\mathbf{T}$  column (bottom). Note that there are two bumps in the  $\mathbf{T}$  activity at internal times that roughly match the stimulus presentation times. The most important feature

of this buffer is that the time of the more recent presentation of the stimulus is more accurately represented than that of the earlier presentation. This can be seen from the fact that the peak around  $\tau^* = -7$  is taller and sharper than the peak around  $\tau^* = -23$ . Thus the value of  $\mathbf{f}$  from a moment in distant past is smeared over many more  $\tau^*$  nodes than the value of  $\mathbf{f}$  from a more recent past moment.

Furthermore, it turns out that the smear is precisely scale invariant. To illustrate this, consider  $\mathbf{f}(\tau)$  to be a Dirac delta function at a moment  $\tau_o$  in the past,  $\mathbf{f}(\tau) = \delta(\tau - \tau_o)$ , and let the present moment be  $\tau = 0$ . Applying eqns. 5 and 6, we obtain

$$\mathbf{T}(0, \tau^*) = \frac{1}{|\tau_o|} \frac{k^{k+1}}{k!} \left( \frac{\tau_o}{\tau^*} \right)^{k+1} e^{-k(\tau_o/\tau^*)} \quad (7)$$

In the above equation both  $\tau_o$  and  $\tau^*$  are negative;  $\mathbf{T}(0, \tau^*)$  is the representation of the delta function input distributed over the set of  $\tau^*$  nodes in the  $\mathbf{T}$  column. The delta function input in real time is represented as a smooth peaked function in the  $\mathbf{T}$  column such that the area underlying this distribution over  $\tau^*$  is always 1, reflecting the area underlying the input function. This can be heuristically verified from fig. 3, where the area under the two bumps are the roughly the same. The term  $1/|\tau_o|$  in the l.h.s of the above equation has the effect of reducing the size of the peak inversely with increasing  $\tau_o$ . The rest of the functional dependence is on the ratio  $(\tau_o/\tau^*)$ , ensuring that the distribution shape linearly scales with  $\tau_o$ . In this sense,  $\mathbf{T}$  represents the history of  $\mathbf{f}(\tau)$  with a scale invariant smear. To quantify how much smear is introduced, we can estimate the width of the peak as the standard deviation  $\sigma$  of  $\mathbf{T}(0, \tau^*)$  from the above equation, which for  $k > 2$  turns out to be

$$\sigma[\mathbf{T}(0, \tau^*)] = \frac{|\tau_o|}{\sqrt{k-2}} \left[ \frac{k}{k-1} \right] \quad (8)$$

The infinitely sharp delta function input to  $\mathbf{f}$  is thus smeared out in  $\mathbf{T}$ , and the width of the smear linearly increases with the time of presentation of the input. Note that  $k$  is the only free parameter here and eq. 8 shows that  $k$  has an inverse influence on the smear: larger the  $k$ —smaller the smear, and smaller the  $k$ —larger the smear. Hence  $k$  can simply be interpreted as the *smear index*. In the limit  $k \rightarrow \infty$ , the smear vanishes and the delta function input propagates into the  $\mathbf{T}$  column exactly as delta function without spreading, as expected in a shift register.

Finally, note that the linearity of eqns. 5 and 6 implies that a linear combination of different functions  $\mathbf{f}$  will lead to a linear combination of representations in  $\mathbf{T}$ . For a more elaborate mathematical description, refer to [14]. In the description so far,  $\tau^*$  is continuous variable. In order to utilize these insights in machine learning applications, we need to construct a buffer with discrete values of  $\tau^*$ . In the following subsections, we discuss several implementation details for constructing a fuzzy buffer from  $\mathbf{T}$ .

### A. Discretized Implementation

Though mathematically convenient, it is not practical to represent all real values of  $\tau^*$  in the  $\mathbf{T}$  column—only discrete values of  $\tau^*$  can be represented and there has to be a minimum  $\tau_{min}^*$  and a maximum  $\tau_{max}^*$ . One can in principle pick any set of  $\tau^*$  values, and this will fix the set of  $s$  values in the  $\mathbf{t}$  column because of the one to one correspondence  $s = -k/\tau^*$ .

The choice of the discrete set of nodes affects the way  $\mathbf{T}$  is constructed from  $\mathbf{t}$ . Note from eq. 6 that the  $\mathbf{L}_k^{-1}$  operator has to take the  $k$ -th derivative of  $\mathbf{t}$  along the  $s$  axis, which will be strongly



affected by the discretization of the  $s$ -axis. However, for any discretized set of  $s$  values, we can appropriately define a discretized derivative that is a linear operator. For notational convenience, let us denote the activity at any moment  $\mathbf{t}(\tau, s)$  as simply  $\mathbf{t}(s)$ . Since  $\mathbf{t}$  is a column vector with the rows labeled by  $s$ , we can construct a derivative matrix  $[D]$  such that

$$\mathbf{t}^{(1)} = [D]\mathbf{t} \quad \implies \quad \mathbf{t}^{(k)} = [D]^k \mathbf{t} \quad (9)$$

The individual elements in the square matrix  $[D]$  depends on the set of  $s$  values. To compute these elements, consider any three successive nodes with  $s$  values  $s_{-1}, s_o, s_1$ . The discretized first derivative of  $\mathbf{t}$  at  $s_o$  is given by

$$\mathbf{t}^{(1)}(s_o) = \frac{\mathbf{t}(s_1) - \mathbf{t}(s_o)}{s_1 - s_o} \left[ \frac{s_o - s_{-1}}{s_1 - s_{-1}} \right] + \frac{\mathbf{t}(s_o) - \mathbf{t}(s_{-1})}{s_o - s_{-1}} \left[ \frac{s_1 - s_o}{s_1 - s_{-1}} \right] \quad (10)$$

The row in  $[D]$  corresponding to  $s_o$  will have non-zero entries only in the columns corresponding to  $s_{-1}, s_o$  and  $s_1$ . These three entries can be read out as coefficients of  $\mathbf{t}(s_{-1})$ ,  $\mathbf{t}(s_o)$  and  $\mathbf{t}(s_1)$  respectively in the r.h.s of the above equation. Thus the entire matrix  $[D]$  can be constructed from any chosen set  $s$  values.

By taking the  $k$ -th power of  $[D]$ , the  $\mathbf{L}_k^{-1}$  operator can be trivially constructed and the activity of the  $\mathbf{T}$  column with the chosen set of  $\tau^*$  values can be calculated at each moment<sup>1</sup>. We have now established that the  $\mathbf{T}$  column activity can be constructed self-consistently from the  $\mathbf{t}$  column for any set of  $\tau^*$  values. From this point onwards, we shall not concern ourselves with intermediate stage  $\mathbf{t}$  column and simply focus on the  $\mathbf{T}$  column.

## B. The optimal choice of nodes distribution

We have shown that the entire history of  $\mathbf{f}$  at any moment is represented in  $\mathbf{T}$  column in a smeared fashion when  $k$  is finite. But the fact that a single delta function input is smeared over many  $\tau^*$  values implies that there is a lot of redundancy in information representation in the  $\mathbf{T}$  column. This information redundancy can be reduced by optimally distributing the nodes along the  $\mathbf{T}$  column. Let  $g(\tau^*)$  represent the number density of nodes along the  $\mathbf{T}$  column. If we number the nodes in the  $\mathbf{T}$  column by  $N$ , ranging from 1 to  $N_{\max}$ , then  $g(\tau^*) \equiv dN/d\tau^*$ . More simply, if successive nodes are  $\tau^*$  and  $\tau^* + \Delta$ , then  $g(\tau^*) = 1/\Delta$ .

In order to construct a truly scale-free buffer, we first note that the redundancy in information representation should be evenly spread over all time scales that are represented in the buffer. This constraint can be formulated in terms of mutual information shared by neighboring nodes in the buffer. In the appendix, we mathematically show that in the presence of scale free input signals, the mutual information shared by any two neighboring buffer nodes can be a constant only if  $g(\tau^*) \propto 1/|\tau^*|$ . This choice of  $g(\tau^*)$  is exactly equivalent to the constraint introduced by eq. 4 in section 2.

Let us now observe a couple of interesting consequences to the choice of  $g(\tau^*) \propto 1/|\tau^*|$ . First note that this choice leads to the following arrangement of nodes in the  $\mathbf{T}$  column.

$$\tau_{min}^*, \tau_{min}^*(1+c), \tau_{min}^*(1+c)^2, \dots, \tau_{min}^*(1+c)^{(N_{\max}-1)} = \tau_{max}^*, \quad (11)$$

and the total number of nodes is

$$N_{\max} = 1 + \frac{\log(\tau_{max}^*/\tau_{min}^*)}{\log(1+c)}. \quad (12)$$

---

<sup>1</sup> To accurately construct the  $k$ -th derivative, we need  $k$  extra nodes in the top and bottom of the  $\mathbf{t}$  column. These  $s$  values are needed in addition to those determined from the one to one correspondence with the chosen  $\tau^*$  values.

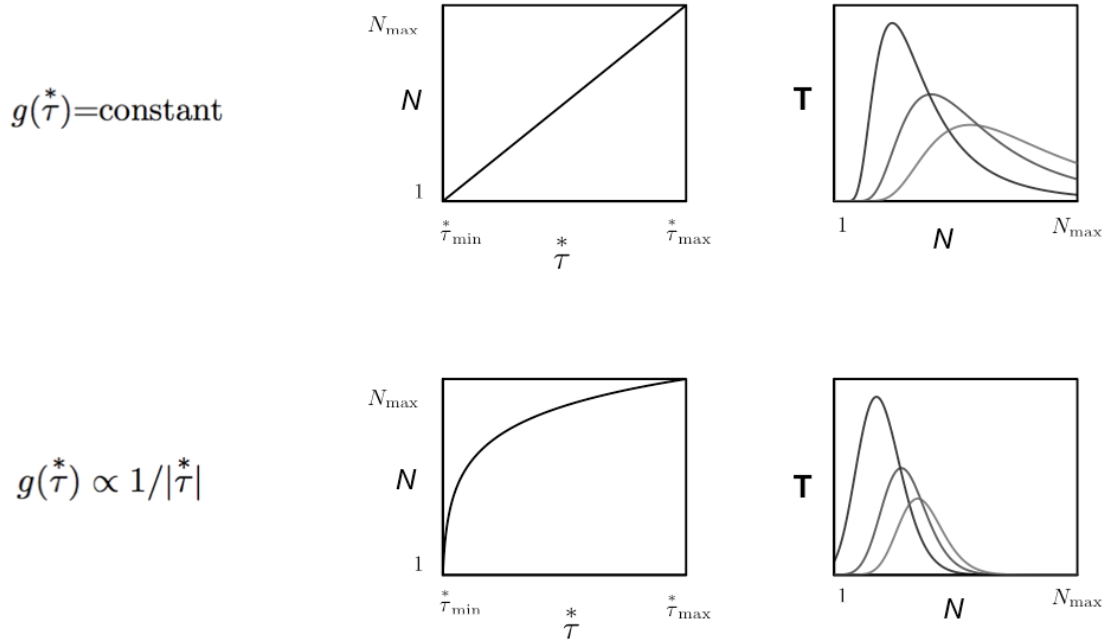


FIG. 4. Pattern of activity across the buffer nodes. The top row corresponds to  $g(\tau^*) = \text{constant}$ , and the bottom row corresponds to  $g(\tau^*) \propto 1/|\tau^*|$ . The left panel plots the ordinal position  $N$  of each buffer node against its  $\tau^*$  value. If a total of  $N_{\max}$  nodes in the buffer are to represent the range from  $\tau_{\min}^*$  to  $\tau_{\max}^*$ ,  $N$  increases linearly with  $\tau^*$  when  $g(\tau^*) = \text{constant}$ , while  $N$  increases logarithmically when  $g(\tau^*) \propto 1/|\tau^*|$ . The  $\mathbf{T}$  activity spread across the different nodes for a delta function input (as in eq. 7) is plotted in the right panel for three different input times in the past. Note that the pattern of activity gets more smeared for larger times when  $g(\tau^*) = \text{constant}$ , but the pattern stays the same with an overall translation and scale reduction when  $g(\tau^*) \propto 1/|\tau^*|$ .

Here the constant  $c$  controls the resolution and denotes the separation between neighboring nodes around  $|\tau^*| = 1$ . In a shift register, the total number of nodes  $N_{\max}$  is proportional to the longest time scale to be represented. But the  $N_{\max}$  in eq. 12 is related to the logarithm of the longest timescale to be represented. This constitutes a tremendous saving of resources—in comparison to a shift register, exponentially larger timescales can be represented in the fuzzy buffer with the same amount of resources.

Another interesting feature of the choice of  $g(\tau^*) \propto 1/|\tau^*|$  is that the pattern of activity across the  $N_{\max}$  nodes is translationally invariant with time—in other words, the smear in the pattern of activity does not increase with time since the input. Recall that the activity across  $\tau^*$  values in response to a delta function input at a time  $\tau_o$  in the past is given by eq. 7, where the smear in the pattern increased proportional to  $|\tau_o|$  (eq. 8). Figure. 4 shows the activity across the  $N_{\max}$  nodes for three values of  $\tau_o$ . When  $g(\tau^*)$  is a constant, the distribution is more smeared for larger  $|\tau_o|$ , but when  $g(\tau^*) \propto 1/|\tau^*|$ , the pattern simply gets translated with an overall scale reduction. It is also clear that the pattern is more symmetric around the peak when  $g(\tau^*) \propto 1/|\tau^*|$ , while it is asymmetric when  $g(\tau^*)$  is a constant. This can be heuristically understood by noting that though the smear along the  $\tau^*$  axis is proportional to the time since the presentation of the delta function, the actual number of nodes representing that timescale is inversely proportional to the timescale

itself when  $g(\tau^*) \propto 1/|\tau^*|$ . Hence when the pattern of  $\mathbf{T}$  activity is plotted w.r.t to the actual nodes  $N$  rather than  $\tau^*$ , the smear is effectively a constant.

It in fact turns out that  $g(\tau^*) \propto 1/|\tau^*|$  is the only choice for which the activity pattern across the nodes is translationally invariant. To see this consider two different values of  $\tau_o$ , say  $\tau_1$  and  $\tau_2$ , in eq. 7, and let us denote the corresponding  $\mathbf{T}$  activities as  $\mathbf{T}_1(0, \tau^*)$  and  $\mathbf{T}_2(0, \tau^*)$  respectively. If we represent the  $\tau^*$  values of the  $N$ -th node in the buffer by  $\tau_N^*$ , then the pattern of activity across the nodes is translationally invariant if and only if  $\mathbf{T}_1(0, \tau_N^*) \propto \mathbf{T}_2(0, \tau_{N+m}^*)$  for some constant integer  $m$ . For this to hold true, we need the quantity

$$\frac{\mathbf{T}_1(0, \tau_N^*)}{\mathbf{T}_2(0, \tau_{N+m}^*)} = \left(\frac{\tau_1}{\tau_2}\right)^k \left[\frac{\tau_{N+m}^*}{\tau_N^*}\right]^{k+1} e^{k\left[\frac{\tau_1}{\tau_N^*} - \frac{\tau_2}{\tau_{N+m}^*}\right]} \quad (13)$$

to be independent of  $N$ . This is possible only when the quantity inside the power law form and the exponential form are separately independent of  $N$ . The power law form can be independent of  $N$  only if  $\tau_N^* \propto (1+c)^N$ , which implies  $g(\tau^*) \propto 1/|\tau^*|$  (see eq. 11). The exponential form is generally dependent on  $N$  except when its argument is zero, which happens if  $(1+c)^m = \tau_2/\tau_1$  for some integer  $m$ . When  $c$  is small compared to 1 and  $\tau_2/\tau_1$  is not very close to 1, there will always exist some integer  $m$  for which the equality will approximately hold. Hence the pattern of activity across the nodes can be translationally invariant only when  $g(\tau^*) \propto 1/|\tau^*|$ . Though this is an interesting feature, we emphasize that the primary reason behind the choice of  $g(\tau^*) \propto 1/|\tau^*|$  is that it equally spreads information redundancy across all time scales as shown in the appendix.

### C. Setting $k$ to minimize information redundancy while avoiding information loss

The choice of  $g(\tau^*) \propto 1/|\tau^*|$  only ensures that the redundancy in information representation introduced due to smearing is equally distributed over buffer nodes. But equal distribution of information redundancy is not sufficient; we would also like to minimize information redundancy. First note that the choice of  $g(\tau^*) \propto 1/|\tau^*|$  does not completely specify the  $\tau^*$  values of the buffer nodes, because  $c$  remains a free parameter in eq. 11. For a given  $c$ , there will be a high information redundancy if  $k$  is too small, while there will be high information loss if  $k$  is too large. Heuristically, if  $k$  is too small for a given  $c$ , then the  $\tau^*$  values of neighboring nodes in the buffer will be sufficiently close so that many nodes will have similar activities in response to an input from the past, resulting in information redundancy. In contrast, if  $k$  is too large for a given  $c$ , the  $\tau^*$  values of neighboring nodes will be sufficiently distant so that the activities of all the nodes could be close to zero for inputs from certain times in the past, resulting in information loss. So we need to appropriately match  $c$  with  $k$  to balance and minimize the information redundancy and the information loss.

The basic idea here is that the information redundancy will be minimal when the information from a single moment in the past is not spread over more than two neighboring nodes. To formalize this, consider a delta function input at a time  $\tau_o$  in the past and let the current moment be  $\tau = 0$ . We shall now look at the activity induced by this input (eq. 7) in four successive buffer nodes,  $N-1$ ,  $N$  and  $N+1$  and  $N+2$ . The  $\tau^*$  values of these nodes are given by eq. 11, for instance  $\tau_N^* = \tau_{min}^*(1+c)^{N-1}$  and  $\tau_{N+1}^* = \tau_{min}^*(1+c)^N$ . From eq. 7, it can be seen that the  $N$ -th node attains its maximum value when  $\tau_o = \tau_N^*$  and the  $N+1$ -th node attains its maximum value when  $\tau_o = \tau_{N+1}^*$ , and for all the intervening times of  $\tau_o$  between  $\tau_N^*$  and  $\tau_{N+1}^*$ , the information about the delta function input will be spread over both  $N$ -th and the  $N+1$ -th nodes. To minimize the information redundancy, we simply require that when  $\tau_o$  is in between  $\tau_N^*$  and  $\tau_{N+1}^*$ , all the nodes other than the  $N$ -th and the  $N+1$ -th nodes should have almost zero activity.

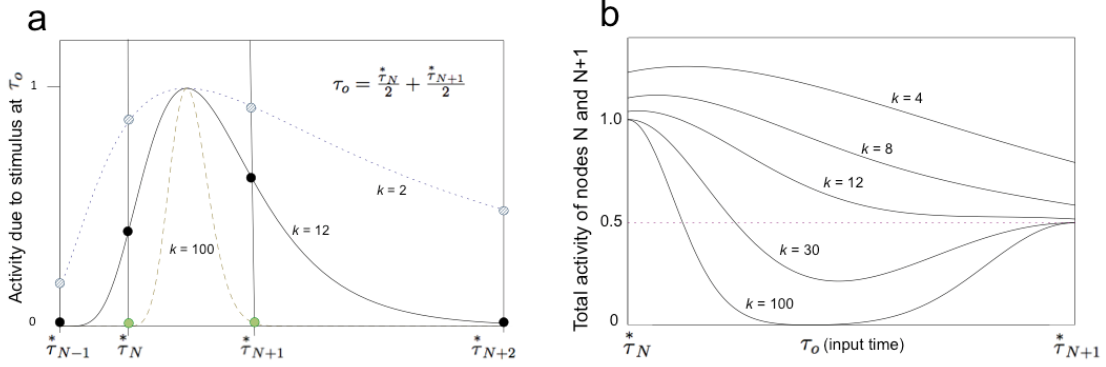


FIG. 5. **a.** The activity of four successive nodes with  $\tau^*$  values given by  $\tau_{N-1}^*$ ,  $\tau_N^*$ ,  $\tau_{N+1}^*$ , and  $\tau_{N+2}^*$  in response to a delta function input at a past moment  $\tau_o = (\tau_N^*/2 + \tau_{N+1}^*)/2$ . The nodes are chosen according to the distribution given by eq. 11 with  $c = 1$ . **b.** The sum of activity of the nodes  $\tau_N^*$  and  $\tau_{N+1}^*$  in response to a delta function input at various times  $\tau_o$  ranging between  $\tau_N^*$  and  $\tau_{N+1}^*$ . For each  $k$ , the activities are normalized to have values in the range of 0 to 1.

Fig. 5a plots the activity for values of  $\tau^*$  between  $\tau_{N-1}^*$  and  $\tau_{N+2}^*$ , with  $c = 1$ , and when  $\tau_o$  is exactly in the middle of  $\tau_N^*$  and  $\tau_{N+1}^*$ . For each value of  $k$ , the activity is normalized so that it lies between 0 and 1. The four vertical lines in fig. 5a represent the 4 nodes and the dots represent the activity of the corresponding nodes. Observe that for  $k = 2$  the activity of all 4 nodes is substantially different from zero, implying a significant information redundancy. At the other extreme, the  $k = 100$  case in fig. 5a, shows that the activity of all the nodes are almost zero, implying that the information about the delta function input at time  $\tau_o = (\tau_N^* + \tau_{N+1}^*)/2$  has been lost. To minimize both the information loss and the information redundancy, the value of  $k$  should be neither too large nor too small. Note that for the  $k = 12$  case in fig. 5a, the activities of the  $N - 1$ -th and the  $N + 2$ -th nodes are almost zero, but activities of the  $N$ -th and  $N + 1$ -th nodes are non-zero.

For any given value of  $c$ , a rough estimate of the appropriate  $k$  can be obtained by matching the difference in the  $\tau^*$  values of the neighboring nodes to the smear  $\sigma$  (the standard deviation as measured by eq. 8) in the distribution over the  $\tau^*$  values.

$$\sigma = \frac{|\tau_{N+1}^*|}{\sqrt{k-2}} \left[ \frac{k}{k-1} \right] \simeq |\tau_{N+1}^* - \tau_N^*| \quad \Rightarrow \quad \frac{k}{(k-1)\sqrt{k-2}} \simeq \frac{c}{1+c}. \quad (14)$$

This condition implies that a large value of  $k$  will be required when  $c$  is small and a small value of  $k$  will be required when  $c$  is large. In particular, note that  $k \simeq 8$  when  $c = 1$ .

We can construct a measure of information loss and use this as an additional constraint. Figure 5b shows the sum of activity of the  $N$ -th and the  $N + 1$ -th nodes for all values of  $\tau_o$  between  $\tau_N^*$  and  $\tau_{N+1}^*$ , with  $c = 1$  for different values of  $k$ . For each  $k$ , the activities are normalized so that the  $N$ -th node attains 1 when  $\tau_o = \tau_N^*$ . Now let us focus on the  $k = 100$  case in fig. 5b. There is a range of  $\tau_o$  values for which the total activity of the two nodes is very close to zero. The input is represented purely by the  $N$ -th node when  $\tau_o$  is close to  $\tau_N^*$ , and is represented purely by the  $N + 1$ -th node when  $\tau_o$  is close to  $\tau_{N+1}^*$ , but at intermediate values of  $\tau_o$  the input is not represented by any node. To avoid such information loss, we shall require that the total activity of the two nodes should not have a local minimum—in other words the minimum should be at the

boundary, at  $\tau_o = \tau_{N+1}^*$ , as seen in figure 5b for  $k = 4, 8$  and  $12$ . For  $c = 1$ , it turns out that there exists a local minimum in the total activity of the two nodes only for values of  $k$  greater than  $12$ . For any given  $c$ , the appropriate value of  $k$  that minimizes the information redundancy and information loss can be estimated by examining a plot similar to fig. 5b with the requirement of absence of local minimum.

In summary, the optimally fuzzy buffer is the set of  $\mathbf{T}$  column nodes with  $\tau^*$  values given by eq. 11, with the value of  $k$  appropriately matched with  $c$  to minimize information redundancy and information loss.

#### IV. UTILITY OF THE FUZZY BUFFER IN TIME SERIES FORECASTING

We have constructed a buffer that satisfies the motivation provided in section 2—optimally sacrificing accuracy to accommodate scale-free fluctuations and enhance the capacity to represent information from very long time scales. Now, with a few simple illustrations we shall compare the performance of the fuzzy buffer to a shift register in time series forecasting. We consider three time series with different properties.

The first was generated by fractionally integrating white noise [10] in a manner similar to that described in section 2. The second and third time series were obtained from the online library at <http://datamarket.com>. The second time series is the mean annual temperature of the Earth from the year 1781 to 1988. The third time series is the monthly average number of sunspots from the year 1749 to 1983 measured from Zurich, Switzerland. These three time series are plotted in the top row of fig. 6. The corresponding two point correlation function of each series is plotted in the middle row of fig. 6. Examination of the two point correlation functions reveal differences between the series. The fractionally-integrated noise series shows long-range correlations falling off like a power law. The temperature series shows correlations near zero (but modestly positive) over short ranges and weak negative correlation over longer times. The sunspots data has both strong positive short-range autocorrelation and a longer range negative correlation, balanced by a periodicity of 130 months corresponding to the 11 year solar cycle.

Our goal here is to illustrate the differences between a simple shift register and the fuzzy buffer. Because our interest is in the effect of representing the time series in the memory buffer and not in the sophistication of the learning algorithm, we use simple linear regression algorithm to learn and forecast these time series.

##### A. Learning and forecasting methodology

Let  $N_{max}$  denote the total number of nodes in the buffer and let  $N$  be an index corresponding to each node ranging from 1 to  $N_{max}$ . We shall denote the value contained in the buffer nodes at any time step  $i$  by  $B_i[N]$ . The time series was sequentially fed into both the shift register and the fuzzy buffer and the buffers were appropriately evolved at each time step. The values in the shift register nodes were shifted downstream at each time step as discussed section 2. At any instant the shift register held information from exactly  $N_{max}$  time steps in the past. The values in the fuzzy buffer were evolved as described in section 3, with  $\tau^*$  values taken to be  $1, 2, 4, 8, 16, 32, \dots, 2^{(N_{max}-1)}$ , conforming to eq. 11 with  $\tau_{min}^* = 1$ ,  $c = 1$  and  $k = 8$ .

At each time step  $i$ , the value from each of the buffer nodes  $B_i[N]$  was recorded along with the value of the time series at that time step, denoted by  $V_i$ . We used a simple linear regression algorithm to extract the intercept  $I$  and the regression coefficients  $R_N$  so that the predicted value

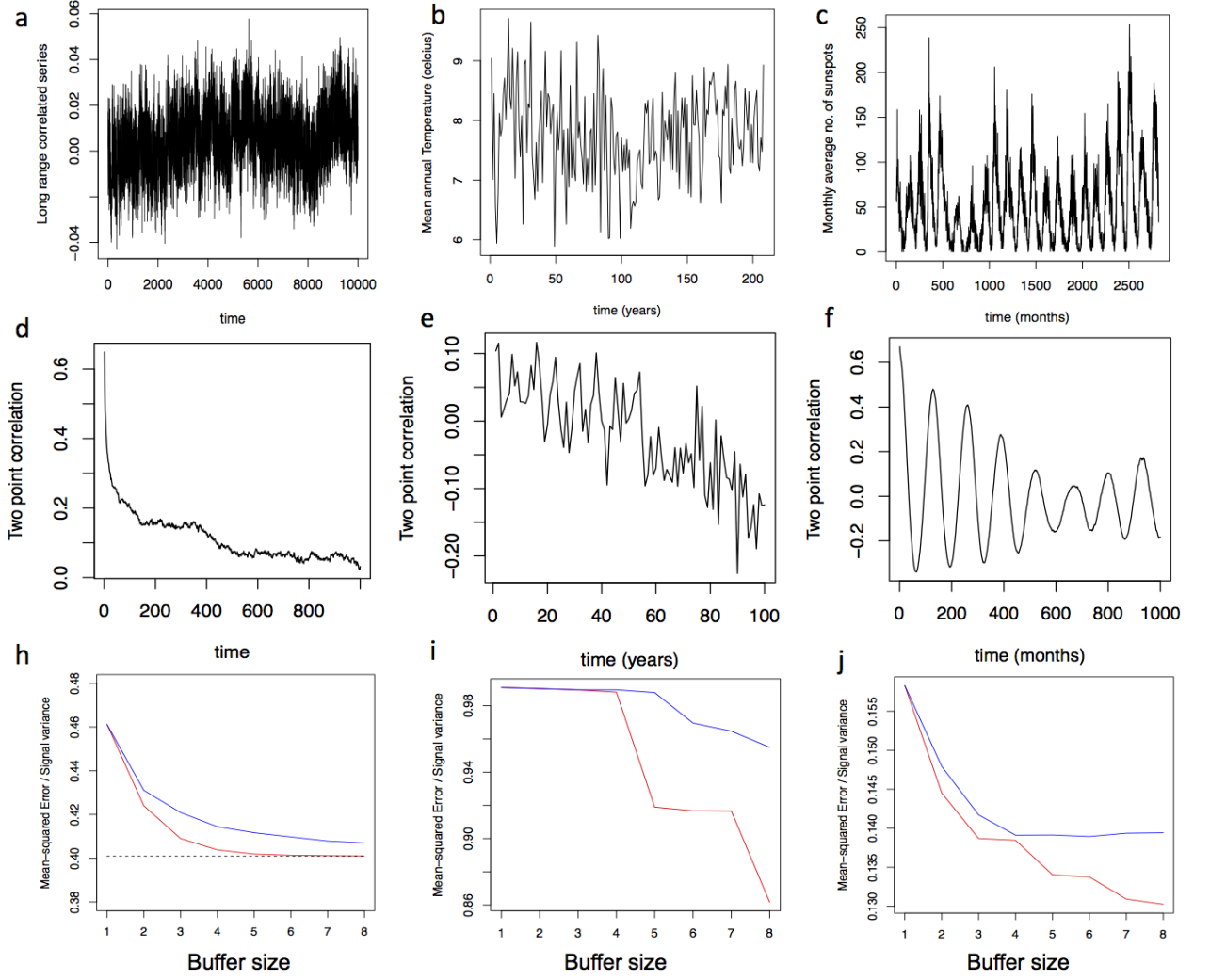


FIG. 6. Time series Forecasting. Top row – **a.** simulated time series with long range correlations based on ARFIMA model with  $d = 0.4$ , and white noise of standard deviation 0.01. **b.** time series of average annual temperature of the Earth from the year 1781 to 1988. **c.** time series of monthly average number of sunspots from the year 1749 to 1983. The middle row shows the two point correlations extracted from each of the time series directly below the time series themselves. The bottom row shows the error in forecasting the corresponding time series in the top row using the fuzzy buffer (red) and using the shift register (blue).

of the time series at each time step  $P_i$  and the squared error in prediction  $E_i$  are

$$P_i = I + \sum_{N=1}^{N_{max}} R_N B_i[N], \quad E_i = [P_i - V_i]^2. \quad (15)$$

The regression coefficients were extracted by minimizing the total squared error  $E = \sum_i E_i$ . For this purpose, we used a standard procedure `lm()` in the open source software R.

The accuracy of forecast is inversely related to the total squared error  $E$ . To get an absolute measure of accuracy we have to factor out the intrinsic variability of the time series. In the bottom row of fig. 6, we plot the mean of the squared error divided by the intrinsic variance in the time series  $var(V_i)$ , for various sizes  $N_{max}$  of the buffer. This quantity would range from 0 to 1; the closer it is to zero, the more accurate the prediction.

*a. Long range correlated series* : The long range correlated series (fig. 6a) is by definition constructed to yield a two point correlation that decays as a power law. This is evident from its two point correlation in fig. 6d that is decaying, but always positive. Since the value of the series at any time step is highly correlated with its value at the previous time step, we can expect to generate a reasonable forecast using a single node buffer that holds the value from the previous time step. This can be seen from fig. 6h, where the error in forecast is only 0.45 with a single buffer node. Adding more buffer nodes reduces the error for both the shift register and the fuzzy buffer. But for a given size of the buffer, the fuzzy buffer always has a lower error than the shift register. This can be seen from fig. 6h where the red curve (corresponding to the fuzzy buffer) is below the blue (corresponding to the shift register).

Since this series is generated by fractionally integrating white noise, the mean squared error cannot in principle be lower than the variance of the white noise used for construction. That is, there is a lower bound for the error that can be achieved in fig. 6h. The dotted line in fig. 6h indicates this bound. Note that the fuzzy buffer approaches this bound with a much smaller number of nodes in the buffer than the shift register.

*b. Temperature series* : The temperature series (fig. 6b) is much more noisy than the long range correlated series, and apparently seems structureless. This can be seen from its small values of its two point correlations in fig. 6e. This is also reflected in the fact that with a small number of buffer nodes, the error is very high. Hence it can be concluded that no reliable short range correlation exist in this series. That is, knowing the average temperature during a given year does not help much in predicting the average temperature of the subsequent year. However, there seems to be a weak negative correlation at longer scales that could be exploited in forecasting. Note from fig. 6i that with additional nodes the fuzzy buffer performs better at forecasting and has a lower error in forecasting than a shift register. This is because the fuzzy buffer can represent much longer timescales than the shift register of equal size, and thereby exploit the long range correlations that exist.

*c. Sunspots series* : The sunspot series (fig. 6c) is less noisy than the other two series considered, and it has an oscillatory structure of about 130 month periodicity. It has high short range correlations, and hence even a buffer with one node that holds the value from the previous time step is sufficient to forecast with an error of only 0.16, as seen in fig. 6j. As before, with more buffer nodes, the fuzzy buffer consistently has a lower error in forecasting than the shift register with equal number of nodes. Note that when the size of the buffer is increased from 4 to 8, the shift register does not improve in accuracy while the fuzzy buffer continues to improve in accuracy.

The previous value in the time series provides most of the information required to predict the next value in the series. With  $c = 1$  (as taken here), the shift register and the fuzzy buffer are precisely the same with only one node. Because most of the variance in the series can be captured by the first node, the difference between the fuzzy buffer and the shift register with additional nodes is not numerically overwhelming when viewed in fig. 6j. However, there is a qualitative difference in the properties of the signal that have been extracted by the two buffers. In order to successfully learn the 130 month periodicity, the information about high positive short range correlations is not sufficient, it is essential to also learn the information about the negative correlations at longer time scales. From fig. 6f, note that the negative correlations exist at a timescale of 50 to 100 months. Hence in order to learn this information, these timescales have to be represented in the buffer. A shift register with 8 nodes cannot represent these timescales but the fuzzy buffer with 8 nodes can.

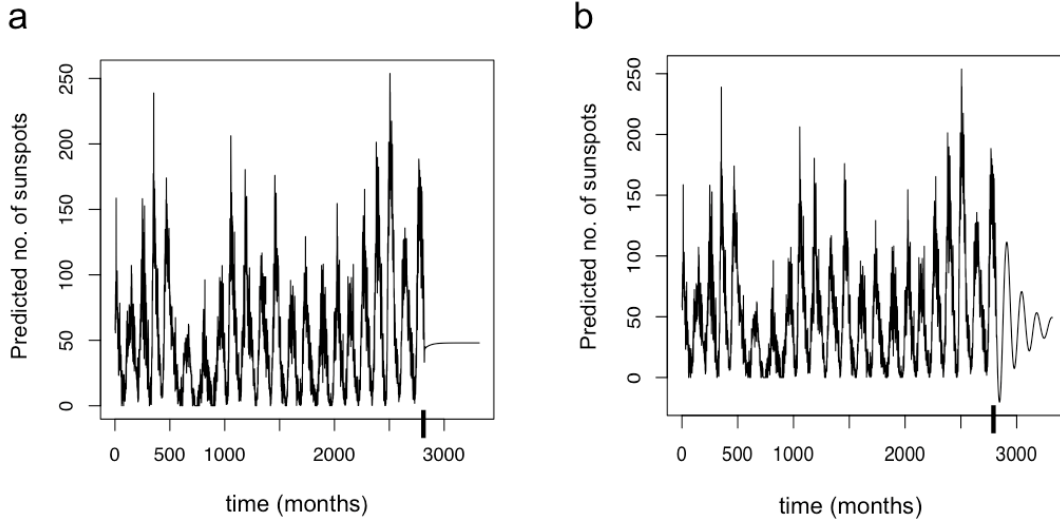


FIG. 7. Forecasting the distant future. The sunspots time series of length 2820 is extrapolated for 500 time steps in the future using a **a.** shift register with 8 nodes, and using the **b.** fuzzy buffer with 8 nodes. The solid tick mark on the  $x$ -axis (at 2820) corresponds to the point where the original series ends and the predicted future series begins.

To illustrate that it is possible to learn the periodicity using the fuzzy buffer, we forecast the distant future values of the series. In figure 7, we extend the sunspots series by predicting it for a future of 500 months. The regression coefficients  $R_N$  and the intercept  $I$  are extracted from the original series of length 2820. For the next 500 time steps, the predictions  $P_i$  are treated as actual values  $V_i$ , and are fed into the buffer to generate the prediction for the next step. Fig. 7a shows the series generated by shift register with 8 nodes. The solid tick mark on the  $x$ -axis at 2820 represents the point at which the original series ends and the predicted future series begins. Note that the series forecasted by the shift register immediately settles on the mean value without oscillation. This is because the time scale at which the oscillations are manifest is not represented by the shift register with 8 nodes. Fig. 7b shows the series generated by the fuzzy buffer with 8 nodes. Note that the series predicted by the fuzzy buffer continues in an oscillating fashion with decreasing amplitude for several cycles eventually settling at the mean value. This is possible because the fuzzy buffer represents the signal at a sufficiently long time scale to capture the negative correlations in the two-point correlation function.

Of course, a shift register with many more nodes can capture the long-range correlations and learn to predict the periodic oscillations in the signal. However the number of nodes necessary to describe the oscillatory nature of the signal needs to be of the order of the periodicity of the oscillation, about 130 in this case. Though it might appear that a shift register with sufficiently large number of nodes is sufficient to such extract all the relevant statistics, note that this would lead to overfitting the data. At least in the case of the simple linear regression algorithm, the number of regression coefficients to be extracted from the data increases with the number of buffer nodes, and extracting a large number of regression coefficients from a finite data set will unquestionably lead to overfitting the data. Hence it would be ideal to use the least number of buffer nodes required to span the relevant time scale, as in the case of the fuzzy buffer.



## V. DISCUSSION

We have demonstrated that in situations where long range correlations are relevant and when the storage resources are finite, the fuzzy buffer is superior to the shift register as a memory system. Over and beyond representing exponentially longer time scales than a shift register, the fuzzy buffer has another useful feature that has not been explicated in the previous section, namely *generalization*. When the time series to be learned is sufficiently long, it is reasonable to assume that the stochasticity in the underlying processes generating the series is statistically well sampled, and the statistics extracted by the learner could indeed correspond to the underlying processes. The accuracy of forecasting hence fundamentally relies on the length of the training series—the number of learning instantiations provided to the learner. However, in real life situations we can be forced to forecast based on very few learning experiences. When there is not sufficient data to extract the statistics of the underlying processes, it is very advantageous to use a fuzzy buffer rather than a shift register. This is because the smeared representation of history in the fuzzy buffer implicitly accounts for the scale free fluctuations in the natural external world. In section 2, we motivated the utility of smearing in the context of a binary valued time series denoting the occurrence of a stimulus. We shall now briefly expand on this to point out that a smeared representation of history in the buffer can speed up learning by facilitating generalization.

Consider a situation in which a delta function stimulus at a time  $\tau$  in the past is followed by some relevant outcome. Let there be a distribution  $p(\tau)$  of  $\tau$  values for which the stimulus yields the outcome. Suppose that a single value of  $\tau$ , say  $\tau_o$ , is chosen from the distribution and the stimulus-outcome sequence is presented to the learner. If the time of occurrence of the stimulus is represented in the learner’s memory without any smear, as in a shift register, then observing the single value  $\tau_o$  does not allow the learner to generalize to other possible values of  $\tau$ . In a shift register, the representation of an event 100 time steps in the past is categorically distinct from the representation of an event 101 time steps in the past. While this property might be optimal in laboratory conditions with precisely timed stimulus-outcome sequence, we would expect that  $p(\tau)$  has some intrinsic spread for naturally occurring events. While we cannot know the properties of  $p(\tau)$  from observing a single value  $\tau_o$ , we can commit to the scalar property based on the ubiquitous scale-free fluctuations in the world. Information about the precise time of the event is smeared out in  $\mathbf{T}$  so that the learner cannot perfectly distinguish  $\tau_o$  from neighboring values. Because the learner cannot perfectly distinguish two events with similar values of  $\tau$ , the response to those two events will also not be perfectly distinguishable. The scalar smear in  $\mathbf{T}$  can hence be seen as an attempt to generalize, that is, attempt to learn the distribution  $p(\tau)$  based on the single observed value  $\tau_o$ .

In this paper, we have argued that a fuzzy buffer offers several advantages over a shift register for representing time-varying information subject to capacity constraints. If this is in fact the case, it seems natural to wonder if such a scale-free fuzzy representation of the past resembles the memory of human and animal learners. After all, animals have evolved in the natural world where predicting the imminent future is crucial for survival. Given the ubiquitous existence of scale-free fluctuations in the natural world, it would have been evolutionarily adaptive for the animals to have developed a memory system that implicitly exploits the existence of such fluctuations. In fact, TILT was developed to account for findings from experimental and cognitive psychology [14]. Numerous behavioral findings from learning and memory as well as timing tasks are consistent with a scale-free representation of past events [19, 20]. In human memory studies, the forgetting curve is usually observed to follow a power law function, which is of course scale-invariant [21, 22]. When humans are asked to reproduce or discriminate time intervals, they exhibit a characteristic scale-invariance in the errors they produce [23, 24]. This is not just a characteristic feature in humans, but in a wide variety of animal species like rats, rabbits and pigeons, as demonstrated

by classical conditioning experiments [25, 26]. These findings seem to suggest that humans and animals might have a memory system that represents the past events in a scale-free fuzzy fashion.

Regardless of whether the fuzzy buffer is a valid model of human memory, we propose that it would be a very useful memory system for an artificial intelligent agent attempting to learn the statistics in real world situations. To emphasize that the utility of the fuzzy buffer goes well beyond the simplified illustrations in this paper, we note the following two points. (i) Invariably, in real situations the agent will have to learn a multi-component (or a vector valued) time series and extract correlational or causal relationships across different components. Though in this paper we have only focused on representing a single component time series in the buffer, it is easy to see that each component of the time series can be represented in a separate fuzzy buffer and the learning algorithm, whatever it may be, can act on these buffers to extract the relevant statistics. (ii) In constructing the fuzzy buffer we simply aimed to optimally represented a stochastic time series with power law two point correlations (see eq. 1). However, the time series to be learned in real life situations would generally have meaningful higher order correlations, especially if the series generator involves non-Gaussian stochasticity. But as long as the series is optimally represented in the memory buffer respecting the leading order statistics (the two point correlations and fluctuations), then the learning algorithm should have a relatively easy task in extracting the higher order correlations. Clearly, the simplified linear regression learning algorithm used in section 5 would have completely ignored any higher order correlations, and that may possibly have led to large inaccuracies in forecasting the temperature series (see fig. 6h); a more sophisticated learning algorithm could have forecasted much better.

In other words, the fuzzy buffer by itself is not a solution to machine learning and artificial intelligence problems; one needs an efficient learning algorithm to act on the fuzzy buffer. On the one hand there exists efficient machine learning algorithms like support vector machines[12, 27] and deep belief networks [13] that learn the statistics by batch processing the entire time series data, requiring the entire time series to be accurately accessible at once. On the other hand there exists incremental learning algorithms in cognitive neuroscience like the adaptive resonance theory [28], that can learn on the fly - in an online fashion without waiting for the entire time series to be available, a feature mimicking human learning. However, there does not exist a learning algorithm designed to work on a fuzzy buffer described as here, and constructing such a learning algorithm would be very fruitful. In short, we propose the fuzzy buffer as a baseline memory representation for statistical learning in general.

## VI. CONCLUSION

Signals with long-range temporal correlations are ubiquitous in the natural world. Such signals present a distinct challenge to machine learners that rely on a shift-register representation of the time series. Here we have described an efficient method for constructing a scale-free representation of temporal history,  $\mathbf{T}$ . The scale-free smear in the  $\mathbf{T}$  representation facilitates the learner to quickly generalize and accommodate for the inherent scale-free temporal fluctuations in the natural world. The optimally fuzzy buffer is constructed by choosing the distribution of nodes that minimizes the information redundancy and information loss, and equally distributes them to all timescales. With a given number of nodes, the fuzzy buffer can represent information from exponentially larger time scales when compared to a shift register. This representation of temporal history may be an extremely useful way to represent time series with long-range correlations for use in machine learning applications.

## APPENDIX: INFORMATION REDUNDANCY ACROSS NODES

Here we quantify information redundancy by deriving explicit expressions for mutual information shared between neighboring buffer nodes. When the input signals are white noise or long-range correlated signals with scale-free two point correlations, we show that equally distributing information redundancy to all scales requires  $g(\tau^*) \propto 1/|\tau^*|$ .

Information about  $\mathbf{f}(\tau)$  is distributed among all the  $\mathbf{T}$  nodes in a smeared fashion. This leads to redundancy in the information represented in these nodes. In order to more clearly understand the ability of the  $\mathbf{T}$  column to represent information regarding  $\mathbf{f}(\tau)$ , we analyze the statistical properties of the  $\mathbf{T}$  column nodes when  $\mathbf{f}(\tau)$  is driven by a stochastic input. Taking the current moment to be  $\tau = 0$ , the activity of a  $\tau^*$  node in the  $\mathbf{T}$  column is given by

$$\mathbf{T}(0, \tau^*) = \int_{-\infty}^0 \frac{1}{|\tau^*|} \left( \frac{\tau'}{\tau^*} \right)^k e^{-k(\frac{\tau'}{\tau^*})} \mathbf{f}(\tau') d\tau' \quad (\text{A1})$$

The expectation value of this node can be calculated by simply averaging over  $\mathbf{f}(\tau')$  inside the integral, which should be a constant if it is generated by a stationary process. By defining  $z = \tau'/\tau^*$ , we find that the expectation of  $\mathbf{T}$  is proportional to the expectation of  $\mathbf{f}$ .

$$\langle \mathbf{T}(0, \tau^*) \rangle = \langle \mathbf{f} \rangle \int_0^\infty z^k e^{-kz} dz \quad (\text{A2})$$

To understand the information representation in terms of correlations among the nodes, we calculate the correlations and mutual information among the  $\mathbf{T}$  nodes when  $\mathbf{f}(\tau)$  is white noise and long-range correlated noise.

### White-noise $\mathbf{f}(\tau)$

Let  $\mathbf{f}(\tau)$  to be white noise, that is  $\langle \mathbf{f} \rangle = 0$  and  $\langle \mathbf{f}(\tau) \mathbf{f}(\tau') \rangle \sim \delta(\tau - \tau')$ . The variance in the activity of each  $\tau^*$  node is then given by

$$\begin{aligned} \langle \mathbf{T}^2(0, \tau^*) \rangle &= \int_{-\infty}^0 \int_{-\infty}^0 \frac{1}{|\tau^*|^2} \left( \frac{\tau}{\tau^*} \right)^k e^{-k(\frac{\tau}{\tau^*})} \left( \frac{\tau'}{\tau^*} \right)^k e^{-k(\frac{\tau'}{\tau^*})} \langle \mathbf{f}(\tau) \mathbf{f}(\tau') \rangle d\tau d\tau' \\ &= \frac{1}{|\tau^*|} \int_0^\infty z^{2k} e^{-2kz} dz \end{aligned} \quad (\text{A3})$$

As expected, the variance of a large  $|\tau^*|$  node is small because the activity in this node is constructed by integrating the input function over a large timescale. This induces an artificial temporal correlation that does not exist in the input function. To see this more clearly, we calculate the time correlation in the activity of a single node,  $\langle \mathbf{T}(\tau, \tau^*) \mathbf{T}(\tau', \tau^*) \rangle$ . With the definition  $\delta = |\tau - \tau'|/|\tau^*|$ , it turns out that

$$\langle \mathbf{T}(\tau, \tau^*) \mathbf{T}(\tau', \tau^*) \rangle = |\tau^*|^{-1} e^{-k\delta} \sum_{r=0}^k \delta^{k-r} \frac{k!}{r!(k-r)!} \int_0^\infty z^{k+r} e^{-2kz} dz \quad (\text{A4})$$

Note that this correlation is nonzero for any  $\delta > 0$ , and it decays exponentially for large  $\delta$ . Hence even a temporally uncorrelated white noise input leads to short range temporal correlations in a  $\tau^*$  node. It is important to emphasize here that such temporal correlations will not be introduced in a shift register. This is because, in a shift register the functional value of  $\mathbf{f}$  at each moment is

just passed on to the downstream nodes in the chain without being integrated, and the temporal autocorrelation in the activity of any node will simply reflect the temporal correlation in the input function.

Let us now consider the instantaneous correlation in the activity of two different nodes. At any instant, the activity of two different nodes in a shift register will be uncorrelated in response to a white noise input. The different nodes in a shift register carry completely different information, making their mutual information zero. But in the  $\mathbf{T}$  column, since the information is smeared across different  $\tau^*$  nodes, the mutual information shared by different nodes is non-zero. The instantaneous correlation between two different nodes  $\tau_1^*$  and  $\tau_2^*$  can be calculated to be

$$\begin{aligned} \langle \mathbf{T}(0, \tau_1^*) \mathbf{T}(0, \tau_2^*) \rangle &= |\tau_2^*|^{-1} \int_0^\infty z^{2k} (\tau_1^*/\tau_2^*)^k e^{-kz(1+\tau_1^*/\tau_2^*)} dz \\ &\propto \frac{(\tau_1^* \tau_2^*)^k}{(|\tau_1^*| + |\tau_2^*|)^{2k+1}} \end{aligned} \quad (\text{A5})$$

The instantaneous correlation in the activity of the two nodes  $\tau_1^*$  and  $\tau_2^*$  is a measure of the mutual information represented by them. Factoring out the individual variances of the two nodes, we have the following measure for the mutual information.

$$\mathcal{I}(\tau_1^*, \tau_2^*) = \frac{\langle \mathbf{T}(0, \tau_1^*) \mathbf{T}(0, \tau_2^*) \rangle}{\sqrt{\langle \mathbf{T}^2(0, \tau_1^*) \rangle \langle \mathbf{T}^2(0, \tau_2^*) \rangle}} \propto \left[ \frac{\sqrt{\tau_1^* \tau_2^*}}{(1 + \tau_1^*/\tau_2^*)} \right]^{2k+1} \quad (\text{A6})$$

This quantity is high when  $\tau_1^*/\tau_2^*$  is close to 1. That is, the mutual information shared between neighboring nodes will be the maximum.

The fact that the mutual information shared by neighboring nodes is non-vanishing implies that there is redundancy in the representation of the information in the set of nodes. Now, to formally apply the principle motivated in section 2, we require that the information redundancy should be equally distributed across all time scales represented by the buffer. Put another way, the redundancy in information representation should be scale-free. Mathematically this can be achieved by setting the mutual information between any two adjacent nodes to be a constant. If  $\tau_1^*$  and  $\tau_2^*$  are any two neighboring nodes, then in order for  $\mathcal{I}(\tau_1^*, \tau_2^*)$  to be a constant,  $\tau_1^*/\tau_2^*$  should be a constant. This can happen only if the  $\tau^*$  values of the nodes are arranged in the form given by eq. 11. In other words, this requirement implies that the density of nodes  $g(\tau^*) \propto 1/|\tau^*|$ .

### Inputs with long range correlations

We will now show that choosing  $g(\tau^*) \propto 1/|\tau^*|$  also equalizes the mutual information between all adjacent nodes even when the input has power-law correlations. Consider the input  $\mathbf{f}(\tau)$  such that  $\langle \mathbf{f}(\tau) \mathbf{f}(\tau') \rangle \sim 1/|\tau - \tau'|^\alpha$  for large values of  $|\tau - \tau'|$ . Reworking the calculations analogous to those leading to eq. A4, we find that the time correlation of a node with itself is

$$\langle \mathbf{T}(\tau, \tau^*) \mathbf{T}(\tau', \tau^*) \rangle = \frac{|\tau^*|^{-\alpha}}{2.4^k} \sum_{r=0}^k C_r \int_{-\infty}^\infty \frac{|v|^{k-r}}{|v + \delta|^\alpha} e^{-k|v|} dv \quad (\text{A7})$$

where  $\delta = |\tau - \tau'|/|\tau^*|$  and  $C_r = \frac{k!(k+r)!}{r!(k-r)!} \frac{2^{k-r}}{(k)^{k+r+1}}$ . The value of  $C_r$  is unimportant, we only need to note that it is a positive number.

When  $\alpha > 1$ , the integral diverges at  $v = -\delta$ , however we are only interested in the case  $\alpha < 1$ . When  $\delta$  is very large, the entire contribution to the integral comes from the region  $|v| \ll \delta$  and the denominator of the integrand can be approximated as  $\delta^\alpha$ . In effect,

$$\langle \mathbf{T}(\tau, \tau^*) \mathbf{T}(\tau', \tau^*) \rangle \sim |\tau|^{-\alpha} \delta^{-\alpha} = |\tau - \tau'|^{-\alpha} \quad (\text{A8})$$

for large  $|\tau - \tau'|$ . The temporal autocorrelation of the activity of any node should exactly reflect the temporal correlations in the input when  $|\tau - \tau'|$  is much larger than the time scale of integration of that node ( $\tau^*$ ). As a point of comparison, it is useful to note that any node in a shift register will also exactly reflect the correlations in the input.

Let us now look at the instantaneous correlations across different nodes. In a shift register, the instantaneous correlation between two nodes  $\tau_1^*$  and  $\tau_2^*$ , will simply be  $|\tau_2^* - \tau_1^*|^{-\alpha}$ . The instantaneous correlation between two nodes in  $\mathbf{T}$  column turns out to be

$$\langle \mathbf{T}(0, \tau_1^*) \mathbf{T}(0, \tau_2^*) \rangle = |\tau_2^*|^{-\alpha} \sum_{r=0}^k X_r \frac{\beta^{k-r} (1 + \beta^{r-\alpha+1})}{(1 + \beta)^{2k-r+1}} \quad (\text{A9})$$

Here  $\beta = |\tau_1^*|/|\tau_2^*|$  and each  $X_r$  is a positive coefficient. By always choosing  $|\tau_2^*| \geq |\tau_1^*|$ , we note the two limiting cases of interest, when  $\beta \ll 1$  and when  $\beta \simeq 1$ .

When  $\beta \ll 1$ , the  $r = k$  term in the summation of the above equation yields the leading term, and the correlation is simply proportional to  $|\tau_2^*|^{-\alpha}$ , which is approximately equal to  $|\tau_2^* - \tau_1^*|^{-\alpha}$ . In this limit where  $|\tau_2^*| \gg |\tau_1^*|$ , the correlation between the two nodes behaves like the correlation between two shift register nodes. When  $\beta \simeq 1$ , note from eq. A9 that the correlation will still be proportional to  $|\tau_2^*|^{-\alpha}$ . Now if  $\tau_1^*$  and  $\tau_2^*$  are neighboring nodes with close enough values, we can evaluate the mutual information between them to be

$$\mathcal{I}(\tau_1^*, \tau_2^*) = \frac{\langle \mathbf{T}(0, \tau_1^*) \mathbf{T}(0, \tau_2^*) \rangle}{\sqrt{\langle \mathbf{T}^2(0, \tau_1^*) \rangle \langle \mathbf{T}^2(0, \tau_2^*) \rangle}} \propto |\tau_2^*/\tau_1^*|^{-\alpha/2}. \quad (\text{A10})$$

Reiterating our requirement from before that the mutual information shared by neighboring nodes at all scales should be the same, we are once again led to choose  $\tau_2^*/\tau_1^*$  to be a constant or equivalently  $g(\tau) \propto 1/|\tau|$ .

## REFERENCES

- 
- [1] S. W. Golomb, *Shift Register Sequences*. Laguna Hills, CA: Aegean Park Press, 1982.
  - [2] B. Mandelbrot, *The Fractal Geometry of Nature*. San Fransisco, CA: W. H. Freeman, 2 ed., 1982.
  - [3] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of Optical Society of America A*, vol. 4, pp. 2379–2394, 1987.
  - [4] A. Torralba and A. Oliva, "Statistics of natural image categories," *Network: Computation in Neural Systems*, vol. 14, no. 3, pp. 391–412, 2003.
  - [5] R. F. Voss and J. Clarke, "1/f noise in music and speech," *Nature*, vol. 258, pp. 317–318, 1975.
  - [6] K. Linkenkaer-Hansen, V. Nikouline, J. M. Palva, and R. J. Ilmoniemi, "Long-range temporal correlations and scaling behavior in human brain oscillations," *Journal of Neuroscience*, vol. 21, pp. 1370–1377, 2001.

- [7] R. T. Baillie, "Long memory processes and fractional integration in econometrics," *Journal of Econometrics*, vol. 73, pp. 5–59, 1996.
- [8] D. L. Gilden, "Cognitive emissions of  $1/f$  noise," *Psychological Review*, vol. 108, pp. 33–56, 2001.
- [9] G. C. Van Orden, J. G. Holden, and M. T. Turvey, "Self organization of cognitive performance," *Journal of Experimental Psychology: General*, vol. 132, pp. 331–350, 2003.
- [10] E. J. Wagenmakers, S. Farrell, and R. Ratcliff, "Estimation and interpretation of  $1/f^\alpha$  noise in human cognition," *Psychonomic Bulletin & Review*, vol. 11, no. 4, pp. 579–615, 2004.
- [11] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality: An explanation of  $1/f$  noise," *Physical Review Letters*, vol. 59, no. 4, pp. 381–384, 1987.
- [12] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [13] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.
- [14] K. H. Shankar and M. W. Howard, "A scale-invariant internal representation of time," *Neural Computation*, vol. 24, pp. 134–193, 2012.
- [15] J. Beran, *Statistics for long-memory processes*. New York: Chapman & Hall, 1994.
- [16] C. W. J. Granger and R. Joyeux, "An introduction to long-memory time series models and fractional differencing," *Journal of Time Series Analysis*, vol. 1, pp. 15–30, 1980.
- [17] J. R. M. Hosking, "Fractional differencing," *Biometrika*, vol. 68, no. 1, pp. 165–176, 1981.
- [18] E. Post, "Generalized differentiation," *Transactions of the American Mathematical Society*, vol. 32, pp. 723–781, 1930.
- [19] P. D. Balsam and C. R. Gallistel, "Temporal maps and informativeness in associative learning," *Trends in Neuroscience*, vol. 32, no. 2, pp. 73–78, 2009.
- [20] C. R. Gallistel and J. Gibbon, "Time, rate, and conditioning," *Psychological Review*, vol. 107, no. 2, pp. 289–344, 2000.
- [21] H. Ebbinghaus, "Memory: A contribution to experimental psychology," New York: Teachers College, Columbia University, 1885/1913.
- [22] C. Donkin and R. M. Nosofsky, "A power-law model of psychological memory strength in short- and long-term recognition," *Psychological Science*, vol. 23, pp. 625–634, 2012.
- [23] B. C. Rakitin, J. Gibbon, T. B. Penny, C. Malapani, S. C. Hinton, and W. H. Meck, "Scalar expectancy theory and peak-interval timing in humans," *Journal of Experimental Psychology: Animal Behavior Processes*, vol. 24, pp. 15–33, 1998.
- [24] J. H. Wearden and H. Lejeune, "Scalar properties in human timing: conformity and violations," *Quarterly Journal of Experimental Psychology*, vol. 61, pp. 569–587, 2008.
- [25] S. Roberts, "Isolation of an internal clock," *Journal of Experimental Psychology: Animal Behavior Processes*, vol. 7, pp. 242–268, 1981.
- [26] M. C. Smith, "CS-US interval and us intensity in classical conditioning of rabbit's nictitating membrane response," *Journal of Comparative and Physiological Psychology*, vol. 3, pp. 679–687, 1968.
- [27] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *Computational Intelligence Magazine, IEEE*, vol. 4 (2), pp. 24–38, 2009.
- [28] S. Grossberg, "How does a brain build a cognitive code?," *Psychological Review*, vol. 87, pp. 1–51, 1980.